

BAB II

TINJAUAN PUSTAKA

2.1. *Front End Website*

Pada kebanyakan *website* rata-rata 80-90% waktu dihabiskan saat memuat sebuah halaman terjadi pada *front end* [10]. Waktu yang dihabiskan pada *front end* sebagian besar digunakan untuk mengunduh *resource* dan memprosesnya [11]. *Resource* yang diunduh paling banyak adalah gambar kemudian berkas JavaScript dan terakhir berkas CSS [12]. Berdasarkan dari fakta ini, Souders merekomendasikan kepada pengembang *website* untuk melakukan optimasi pada sisi *front end* terlebih dahulu [7].

Sudah banyak penelitian sebelumnya yang membahas mengenai optimasi pada sisi *front end website*. Hasil yang didapati dari optimasi yang dilakukan pada penelitian sebelumnya adalah mengurangi ukuran berkas [13] [3] [14], mengurangi *request file* [13], mengurangi *bandwidth* [11] [15], dan mengurangi waktu muat halaman [3] [15].

2.2. Optimasi JavaScript

Penggunaan JavaScript hampir selalu ada pada sebuah *website*. Menurut [16], 97,1% *website* menggunakan JavaScript. JavaScript tidak hanya berperan hanya untuk memanipulasi DOM HTML, tetapi juga digunakan untuk menjadikan *website* lebih interaktif dan jembatan antara browser dan server dan lain sebagainya [14]. Rata-rata ukuran berkas JavaScript adalah kedua terbesar setelah gambar dari keseluruhan ukuran halaman *website* [12]. Untuk itu optimasi JavaScript masih relevan dilakukan karena *website* hampir tidak lepas dari penggunaan JavaScript.

Optimasi JavaScript yang dilakukan pada penelitian ini adalah teknik *minify*. Penelitian yang dilakukan Sakamoto[14], *minify* JavaScript dapat mengurangi 39% ukuran berkas JavaScript dan pada penelitian yang dilakukan Himawan[3], *minify* JavaScript mengurangi ukuran berkas sampai 53.85%.

2.3. Optimasi CSS

Penggunaan CSS (*Cascading Style Sheets*) hampir tidak dapat dipisahkan dengan tampilan halaman *website*, CSS sulit tergantikan karena kemudahan dalam mempercantik tampilan *website* [3]. Tanpa CSS, *website* tidak akan memiliki tampilan yang *responsive*. Namun, CSS dapat menimbulkan masalah performa

mulai dari ukuran berkas yang membengkak hingga *blocking* [12]. Pada penelitian ini, teknik optimasi CSS yang digunakan adalah teknik *minify* karena pada penelitian yang dilakukan Himawan [3], *minify* CSS terbukti dapat mengurangi ukuran berkas CSS sampai 52.71% dari ukuran sebelum dioptimasi.

2.4. Optimasi Gambar

Gambar menjadi hal yang sangat penting dan harus ada pada sebuah *website*. Mengapa gambar harus ada dalam sebuah halaman *website* untuk mendukung isi konten yang ditampilkan? Karena pengguna tidak lagi hanya ingin berselancar di internet, namun juga ingin mendapatkan kesan [17]. Gambar yang relevan dapat memengaruhi keterlibatan, penjualan, dan pengalaman pengguna [17].

Tabel 2.1 Ukuran Halaman Website Berdasarkan Tipe Berkas

Persentil	Total (KB)	HTML (KB)	JS (KB)	CSS (KB)	Gambar (KB)	Dokumen (KB)
90	6945	110	1131	240	5220	52
75	3774	58	721	129	2434	26
50	1934	27	391	62	983	14
25	924	12	186	26	319	8
10	397	6	76	8	78	4

Tabel 2.1. merupakan statistik yang dikutip dari Web Almanac 2019 [12]. Gambar merupakan tipe berkas yang paling besar dan paling banyak dimuat pada *website*. Pada persentil ke 90, total ukuran gambar mencapai 5.2 MB dari 7 MB total ukuran halaman atau dengan kata lain 74% dari ukuran keseluruhan halaman *website* adalah gambar. Maka dapat disimpulkan bahwa jika gambar tidak dioptimasi dapat menjadi penyebab membengkaknya ukuran dan akan mempengaruhi performa halaman *website*.

Teknik optimasi gambar yang digunakan pada penelitian ini antara lain:

a. Teknik *Lazy Loading*

Pada penelitian yang dilakukan Nurshuhada *et al* [18], *website* yang menerapkan teknik *Lazy Loading* mendapatkan angka *benchmark* performa yang lebih tinggi dibandingkan dengan yang tidak menerapkan teknik *Lazy Loading*. Selain itu, manfaat dari *Lazy loading* untuk pengguna *mobile* adalah mengurangi *bandwidth* dan mempercepat waktu *loading* halaman *website* [19].

b. Menerapkan gambar yang *responsive*

Gambar berukuran besar yang dimaksudkan untuk perangkat *desktop* seharusnya tidak ditampilkan pada *website* yang diakses dari perangkat berlayar kecil seperti perangkat *mobile*. Pengguna *mobile* tidak ingin membuang *bandwidth* dengan mengunduh gambar besar yang dimaksudkan untuk perangkat *desktop* [20]. Begitupun sebaliknya, jika gambar yang ditampilkan terlalu kecil untuk perangkat *desktop*, maka gambar akan terlihat berbintik.

Idealnya, sebuah *website* memiliki beberapa gambar dengan resolusi yang berbeda sesuai dengan perangkat yang mengaksesnya. Selain untuk mengurangi *bandwidth*, dengan memiliki beberapa gambar dengan resolusi yang berbeda akan mempercepat proses dan waktu muat sebuah halaman *website* terutama saat diakses pada perangkat *mobile*.

c. Konversi atau merubah format gambar

Gambar bergerak (GIF) masih disukai oleh pengguna, namun dari segi ukuran berkas, video masih lebih kecil dibandingkan GIF. Video dengan kualitas serupa dengan GIF cenderung berukuran jauh lebih kecil [21]. Pada penelitian ini akan mengukur perbandingan ukuran GIF yang dirubah menjadi video dan efek terhadap kecepatan akses halaman *website*.

Selain merubah GIF menjadi video, merubah format gambar menjadi JPEG atau WebP dapat mengurangi ukuran berkas tanpa berdampak pada pengalaman visual [12]. Rasio Kompresi gambar WebP lebih baik dibandingkan gambar yang berformat JPEG dan PNG [22].

Untuk itu pada penelitian ini akan melakukan pengujian *website* yang menggunakan gambar berformat WebP dan PNG.

2.5. Penggunaan SPA

Website yang menerapkan *Single Page Application* (SPA) dapat memberikan banyak keuntungan daripada *website* yang masih menggunakan arsitektur pemuatan konten *Multiple Page Application* (MPA). Penerapan SPA akan memberikan pengalaman pengguna yang lebih baik karena lebih cepat daripada *website* MPA dikarenakan setiap kali pengguna ingin berpindah halaman harus berkomunikasi dengan server [23]. Selain mengatasi permasalahan kecepatan, SPA terbukti mengatasi permasalahan penggunaan *bandwidth* yang boros [15].

